# Optimum Velocity Profile of Multiple Bernstein-Bézier Curves Subject to Constraints for Mobile Robots

ANDREJ ZDEŠAR, Faculty of Electrical Engineering
IGOR ŠKRJANC, Faculty of Electrical Engineering

This paper deals with trajectory planning that is suitable for nonholonomic differentially driven wheeled mobile robots. The path is approximated with a spline which consist of multiple Bernstein-Bézier curves that are merged together in a way that continuous curvature of the spline is achieved. The paper presents the approach for optimization of velocity profile of Bernstein-Bézier spline subject to velocity and acceleration constraints. For the purpose of optimization velocity and turning points are introduced. Based on these singularity points local segments are defined where local velocity profiles are optimized independently of each other. From the locally optimum velocity profiles the global optimum velocity profile is determined. Since each local velocity profile can be evaluated independently, the algorithm is suitable for concurrent implementation and modification of one part of the curve does not require recalculation of all local velocity profiles. These properties enable efficient implementation of the optimization algorithm. The optimization algorithm is also suitable for the splines that consist of Bernstein-Bézier curves that have substantially different lengths. The proposed optimization approach was experimentally evaluated and validated in simulation environment and on real mobile robots.

CCS Concepts: • **Computing methodologies** → **Planning and scheduling**; **Robotic planning**; **Control methods**; **Motion path planning**;

Additional Key Words and Phrases: Mobile robots, parametric curves, path planning, trajectory optimization, velocity profile

**ACM Reference format:**
Andrej Zdešar and Igor Škrjanc. 2018. Optimum Velocity Profile of Multiple Bernstein-Bézier Curves Subject to Constraints for Mobile Robots. *ACM Trans. Intell. Syst. Technol.* 1, 1, Article 1 (January 2018), 24 pages.
https://doi.org/10.1145/3183891

## 1 INTRODUCTION

Autonomous mobile robots are becoming an important part of the highly automated factories and they are also slowly penetrating into our homes and streets. These systems need to be capable of autonomous task planning and action scheduling. Various control and action planning approaches have been developed during robot soccer competitions [2, 20, 24, 27]. From the kinematical aspect, autonomous robots should be able to follow the path, which is planned to satisfy a certain given criteria [12, 36, 37, 46], they should avoid obstacles [12, 16, 21, 39], they should prevent collisions between themselves [13, 19, 49] and of course they should be controlled to follow the path from the initial to the target location. This paper deals with trajectory planning, which is one of the essential problems in autonomous mobile robotics. A lot of effort has been put to this problem in


2157-6904/2018/1-ART1 $15.00
https://doi.org/10.1145/3183891

past years in the sense of different optimal solutions. The trajectory could be optimal in time needed from the start to the target point [18, 32, 40, 50, 53], it could be optimal in minimum length [54], minimum energy consumption [28, 38, 47] and many others [4, 11, 31, 36, 37, 49]. The path planning methods can provide the solutions for static or dynamic and known or unknown environment [14, 23, 41, 42, 55]. The approaches such as potential fields methods [3, 22, 52], the real-time limit-cycle navigation method [27], the path planning using harmonic function-based probabilistic road maps [25], the methods based on graph theory such as Voronoi diagrams [9, 19, 44, 51], and variations of D$^\star$ algorithm [6, 10, 15], have been proposed for the path planning. In the literature an important part of path planning problems methodologically appear in the field of intelligent systems [1, 7, 20, 26, 34, 35, 43, 45, 48, 52]. An important area of path planning methods is based on geometrical curves in polynomial way [17, 33, 39], and especially Bernstein-Bézier curves [8, 19, 24, 29, 30, 32, 49], which are also considered in this paper, since these curves have some nice properties that can be leveraged in the optimization of the path (given in Section 3).

Trajectory planning can be broken down into two steps: path searching (shape optimization) and velocity profile optimization (optimization of the time function). The velocity profile path optimization, which is the main topic of this paper, has been considered in [32, 40, 47, 50]. In this paper, time optimal velocity profile planning subject to acceleration and velocity constrains is presented. We present the algorithm for velocity profile optimization that is based on a set of turning and velocity points (singularity points). Around each turning point locally optimum velocity profiles are defined. Based on the set of singularity points, the sufficient domain of locally optimum velocity profiles is defined in a way that the local velocity profiles each can be evaluated independently of each other. The locally optimum velocity profiles can be combined into a globally optimum velocity profile. In the paper we present how the optimization approach can be used on a parametric spline that consist of multiple Bernstein-Bězier curves. It turns out that the entire trajectory need to have continuous curvature, which can be achieved with merging multiple Bernstein-Bězier curves using appropriate merging conditions.

The trajectory optimization approach presented in the paper can be used in various applications of differentially driven wheeled mobile robots. An example application of the case considered is the search for velocity profile of a delivery robot in a factory that drives along a fixed track between different delivery stations in order to minimize the delivery times. In this case it is assumed that the shape of the path is given in advance and should not be changed in the optimization. Therefore this optimization problem also does not consider dynamic accommodation of the path to the current situation in the environment (e.g. other delivery robots). Another example of application of trajectory optimization is in the case of the cooperation of a wheeled mobile system with a human, e.g. as for example in the rehabilitation therapy or autonomous wheel chairs inside buildings. Also in this case the shape of the path can be prescribed and the goal is to find the optimum velocity profile that respects the acceleration and velocity constraints. In this case the constraints are defined based on the patient capabilities or based on the desired drive comfort and health and safety limitations. In any case the acceleration and velocity constraints are upper bounded by the maximum allowable dynamic properties of the wheeled mobile system that insure normal drive conditions (e.g. no wheel slippage, no turn-over or loosing ground contact with any of the wheels).

The paper is organized as follows. In Section 2 the problem is stated and main contributions of the paper are presented. The concept of path planning is described in Section 3. The experimental results of the time optimal path planning with different constraints are given in Section 4. And finally, in Section 5 the conclusions are given.

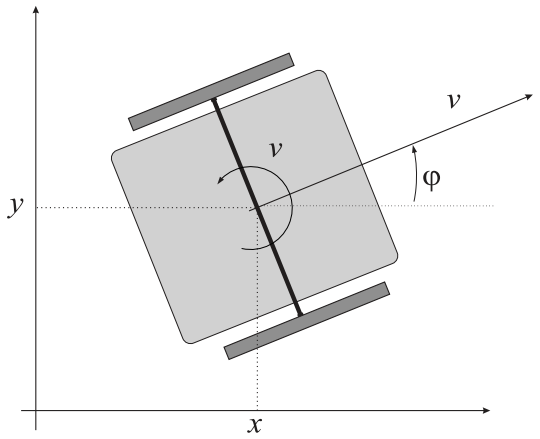


Fig. 1. The generalized coordinates of the mobile robot.

## 2 STATEMENT OF THE PROBLEM

The time optimal design of Bernstein-Bézier curves is proposed for nonholonomic mobile system in a two dimensional free-space environment. We have in mind a two-wheel differentially driven mobile robot. The architecture of the robot has a non-integrable constraint in the form $\dot{x}\sin\varphi - \dot{y}\cos\varphi = 0$ resulting from the assumption that the robot cannot slip in the lateral direction, where $\boldsymbol{q}^T(t) = [x(t)\ y(t)\ \varphi(t)]$ are the generalized coordinates as defined in Figure 1. The kinematic model of the mobile robot with tangential velocity $v(t)$ and angular velocity $\omega(t)$ is

$$\dot{\boldsymbol{q}}(t) = \begin{bmatrix} \cos\varphi(t) & 0 \\ \sin\varphi(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} . \tag{1}$$

A path can be defined as a parametric curve $\boldsymbol{r}_\lambda^T(\lambda) = [x_\lambda(\lambda),\ y_\lambda(\lambda)]$, where the parameter $\lambda$ is defined in the range $0 \le \lambda \le 1$. The parameter $\lambda$ is known as normalized time and its relation to the real time $t$ is considered to be non-linear, i.e. $\lambda = \lambda(t)$, in a way that the condition $\boldsymbol{r}_\lambda(\lambda(t)) = \boldsymbol{r}(t)$ is satisfied, i.e. the shape of the curve is not changed. Sometimes a pure linear relation between the normalized and real time is considered ($\lambda(t) \propto t$), but not in this paper.

From the equivalence of the infinitesimals of the curve with respect to the normalized and real time:

$$\mathrm{d}r_\lambda(\lambda(t)) = \mathrm{d}r(t) \tag{2}$$

$$v_\lambda(\lambda(t))\,\mathrm{d}\lambda = v(t)\,\mathrm{d}t\,, \tag{3}$$

the relation between the tangential velocity $v(t)$ with respect to the real time and the tangential velocity $v_\lambda(\lambda)$ with respect to the normalized time is obtained:

$$v(t) = v_\lambda(\lambda(t))\dot{\lambda}(t) = \sqrt{x_\lambda'^{\,2}(\lambda(t)) + y_\lambda'^{\,2}(\lambda(t))}\,\dot{\lambda}(t)\,. \tag{4}$$

The derivatives with respect to the normalized time are marked with an apostrophe (e.g. $x_\lambda'(\lambda) = \frac{\mathrm{d}x_\lambda(\lambda)}{\mathrm{d}\lambda}$) and derivatives with respect to the real time are marked with a dot (e.g. $\dot{x}(t) = \frac{\mathrm{d}x(t)}{\mathrm{d}t}$). In a similar way as relation between the tangential velocity with respect to real time $t$ and normalized

time $\lambda$ is given in (4), the following entities can be derived: curve direction $\varphi(t)$,

$$\varphi(t) = \arctan \frac{y'_\lambda(\lambda(t))}{x'_\lambda(\lambda(t))} + C\pi \; = \varphi_\lambda(\lambda(t)), \quad C \in \mathbb{Z}, \tag{5}$$

angular velocity $\omega(t)$,

$$\omega(t) = \frac{x'_\lambda(\lambda(t))y''_\lambda(\lambda(t)) - x''_\lambda(\lambda(t))y'_\lambda(\lambda(t))}{v^2_\lambda(\lambda(t))}\dot{\lambda}(t) = \omega_\lambda(\lambda(t))\dot{\lambda}(t)\,, \tag{6}$$

tangential acceleration $a_T(t)$,

$$\begin{aligned} a_T(t) &= \frac{x'_\lambda(\lambda(t))x''_\lambda(\lambda(t)) + y'_\lambda(\lambda(t))y''_\lambda(\lambda(t))}{v_\lambda(\lambda(t))}\dot{\lambda}^2(t) + v_\lambda(\lambda(t))\ddot{\lambda}(t) = \\ &= a_{T\lambda}(\lambda(t))\dot{\lambda}^2(t) + v_\lambda(\lambda(t))\ddot{\lambda}(t) \end{aligned} \tag{7}$$

and radial acceleration $a_R(t)$,

$$a_R(t) = \omega_\lambda(\lambda(t))v_\lambda(\lambda(t)) = a_{R\lambda}(\lambda(t))\dot{\lambda}^2(t)\,. \tag{8}$$

The curve curvature $\kappa(t)$ is defined as

$$\kappa(t) = \frac{\omega_\lambda(\lambda(t))}{v_\lambda(\lambda(t))} = \kappa_\lambda(\lambda(t))\,. \tag{9}$$

The relation between real and normalized entities in (4) to (9) are dependent on the time-normalization function $\lambda(t)$, except for the direction (5) and curvature (9) of the curve. It is easy to observe that these relations simplify if only a simple linear relation between the normalized and true time is considered. But when a general non-linear mapping $\lambda(t)$ is considered, the curve velocity profile can be designed independently of the curve shape.

The idea in this paper is to find a transparent algorithm to define the maximal allowed velocity profile on Bernstein-Bézier curve to stay inside its acceleration and velocities constraints. Those constraints are given with the maximal radial acceleration $a_{R_{max}}$, the maximal tangential acceleration $a_{T_{max}}$, which can be defined based on the dynamical constraints of the mobile robot (e.g. based on the wheel friction, maximal allowable centripetal force) [32]. Additionally, the tangential and radial acceleration should be confined to the inside of the ellipsoid

$$\left(\frac{a_T}{a_{T_{max}}}\right)^2 + \left(\frac{a_R}{a_{R_{max}}}\right)^2 \leq 1\,. \tag{10}$$

The problem is also constrained with the maximal tangential velocity $v_{max}$ and the maximal angular velocity $\omega_{max}$ of the mobile robot. These maximum possible velocities are determined by the capabilities of the robot actuators and also the environmental conditions (e.g. type of the surface). However, in some practical applications these velocities could be constrained even more. For example, these velocities could be bounded based on the desired maximum allowable or stopping time or stopping distance, given known maximum deceleration of the robot (e.g. $v_{max} = \sqrt{2a_{T_{max}}d_{min}}$, where $d_{min}$ is the minimum stopping distance). It is also assumed that the mobile robot can only move forward. This means that the tangential velocity is always positive ($v \geq 0$). This is a reasonable assumption that is not uncommon in the field of autonomous mobile robots [5]. Even though robot drive mechanics almost always allow driving in reverse direction, this is not always desired, since many mobile robots only have sensors that can detect obstacles in the front part of the robot (e.g. due to sensor cost and also mounting restrictions). Therefore safe and collision-free robot operation can not be guaranteed for reverse driving.

Given a feasible path from start to goal point, the optimization problem is to find the velocity profile that reaches the end of the path in minimum time in a way that none of the acceleration and

velocity constraints are violated. Needless to say that the velocity profile need not only be optimal but also feasible.

The velocity profile optimization presented in this paper is designed for use on a spline that consists of multiple Bézier curves, which may not all be of the same order. To be able to satisfy the acceleration constraints, the final tangential and angular velocity profiles need to be continuous. According to (9) it is therefore required that the curvature of the entire curve is also continuous. Therefore, for the sake of completeness, the rules for joining multiple Bézier curves of different orders in a way that the desired continuity of the curve curvature is achieved are also presented.

The proposed velocity profile optimization approach is based on a set of singularity (velocity and turning) points, which will be defined later. It turns out that in the case of continuous curvature the complete set of singularity points can be separated into velocity and turning points if only class of a single element in the set of singularity points is known.

In the presented approach only the curve velocity profile is optimized, i.e. we are only searching for the optimum time function $\lambda(t)$ that ensures the fastest traversal of the trajectory. The shape of the curve is not and should not be modified during the optimization process. Therefore the shape of the curve can also be considered as an additional constraint that needs to be satisfied. However, due to approximations and numerical artifacts some small deviations from the desired shape of the path are allowed — the error is also allowed to slowly drift with time. If this error is not too large, it can be eliminated with an appropriate feedback control without (or appropriately small) violation of the velocity and acceleration constraints. The proposed approach is also suitable for parallelization, since the optimum velocity profile can be determined based on independent optimization of local velocity profiles that are bounded by singularity (velocity and turning) points. This is important in the case of time-critical operations as it is the case of on-line path generation or optimization of path shape along the velocity profile. If the shape of the path also need to be optimized, the presented velocity profile optimization algorithm can be used as a subroutine to find the optimum trajectory in a similar way as it was presented in [47, 50].

## 3 OPTIMUM VELOCITY PROFILE

An optimization of the velocity profile is made on a spline that consist of multiple Bernstein-Bézier curves, which can be used to approximate an arbitrary path with the desired smoothness. First, Bernstein-Bézier parametric curves are introduced briefly in Section 3.1. In Section 3.2 the approach for merging multiple Bézier curves into a single spline in a way that the spline curvature, tangential velocity and angular velocity are continuous functions are presented. Then the velocity and turning points are introduced (Section 3.3), which are used in the proposed approach (Section 3.4) of velocity profile optimization of the spline subject to velocity and acceleration constraints that are declared in Section 2.

### 3.1 Bernstein-Bézier parametric curves

Given a set of $b \in \mathbb{N}$ control points $P_0, P_1, \ldots, P_b$, the corresponding Bernstein-Bézier curve (or Bézier curve) is given by

$$\boldsymbol{r}_\lambda(\lambda) = \sum_{i=0}^{b} \boldsymbol{p}_i B_{i,b}(\lambda)\,, \tag{11}$$

where $B_{i,b}(\lambda)$ is a Bernstein polynomial, $\lambda$ is a normalized time variable ($0 \le \lambda \le 1$) and $\boldsymbol{p}_i$, $i = 0, 1, \ldots, b$, stands for the local vector of the control point $P_i$. Bézier curves can be defined for $N$-dimensional space, $N \in \mathbb{N}$. In the case of a 2-D space the curve $\boldsymbol{r}_\lambda(\lambda)$ and the vectors $\boldsymbol{p}_i$ are two element vectors: $\boldsymbol{r}_\lambda^T(\lambda) = [x_\lambda(\lambda),\ y_\lambda(\lambda)]$ and $\boldsymbol{p}_i^T = [p_{x,i},\ p_{y,i}]$. The Bernstein polynomials, which

are the base functions in the Bézier-curve expansion, are given as:

$$B_{i,b}(\lambda) = \binom{b}{i}\lambda^i(1-\lambda)^{b-i}, \quad i = 0, 1, \ldots, b, \tag{12}$$

which have the following properties (the domain of definition is $0 \leq \lambda \leq 1$): $0 \leq B_{i,b}(\lambda) \leq 1$ for every $i = 0, 1, \ldots, b$ and $\sum_{i=0}^{b} B_{i,b}(\lambda) = 1$. An example of a Bézier curve with eight control points (seventh-order curve, $b = 7$) is shown in Figure 2.

The derivative of a Bézier curve is again a Bézier curve:

$$\frac{\mathrm{d}\boldsymbol{r}_\lambda(\lambda)}{\mathrm{d}\lambda} = b\sum_{i=0}^{b-1} B_{i,b-1}(\lambda)(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i). \tag{13}$$

The derivative of order $d \geq 0$ at the end points is dependent only on the first or last $d$ control points, respectively.

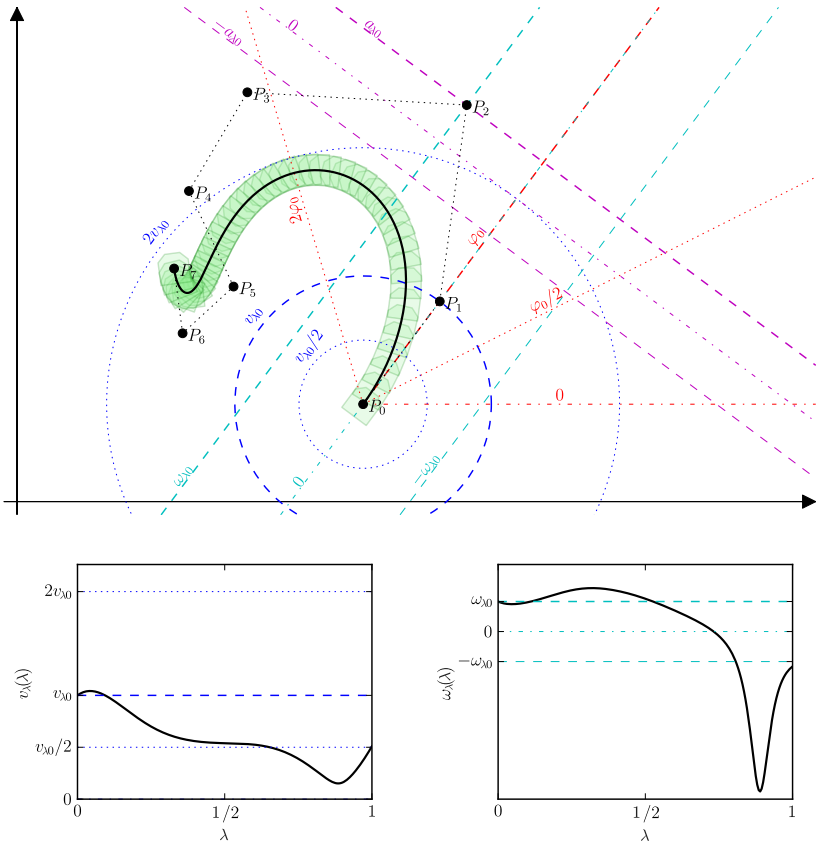


Fig. 2. Example of a Bézier curve with eight control points $\boldsymbol{p}_0$ to $\boldsymbol{p}_7$ and a graphical representation of the influence of the control points $\boldsymbol{p}_0$, $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ on properties of the curve in the start point of the curve ($v_{\lambda 0} \neq 0$). At the bottom normalized tangential (left) and angular (right) velocity profiles of the curve are shown.

Entities (4), (5), (6) and (9) can be expressed with control points if the terms in the equations are substituted with the appropriate derivatives. We are especially interested in these propertied in the curve end points, since the values in the end points need to be considered in combining the curves

into a spline — the smoothness of the spline depends on the conditions considered in the end points. For example, the tangential velocity in the starting point of the curve can be expressed as

$$v_{\lambda 0} = \lim_{\lambda \to 0^+} v_\lambda(\lambda) = b\sqrt{(p_{x,1} - p_{x,0})^2 + (p_{y,1} - p_{y,0})^2}\,, \tag{14}$$

and the direction of the curve in the starting point is

$$\varphi_0 = \lim_{\lambda \to 0^+} \varphi_\lambda(\lambda) = \arctan\frac{p_{y,1} - p_{y,0}}{p_{x,1} - p_{x,0}} + C\pi\,, \quad C \in \mathbb{Z}\,. \tag{15}$$

However, definition (15) is only valid if the velocity in the starting point is not zero, i.e. $\boldsymbol{p}_0 \neq \boldsymbol{p}_1$. If $\boldsymbol{p}_0 = \boldsymbol{p}_1$ the direction of the curve in the starting point can be defined as

$$\varphi_0 = \lim_{\lambda \to 0^+} \varphi_\lambda(\lambda) = \arctan\frac{p_{y,2} - p_{y,0}}{p_{x,2} - p_{x,0}} + C\pi\,, \quad C \in \mathbb{Z}\,. \tag{16}$$

Therefore some special care need to be taken when the tangential velocity (and derivatives of the tangential velocity up to some higher derivative) is zero. Normally we assume that the tangential velocity along the curve is non-zero to avoid these singularities.

The following equations show how the control points should be placed in order to achieve some desired properties in the starting point of the curve ($v_{\lambda 0} \neq 0$):

$$\boldsymbol{p}_0 = \begin{bmatrix} x_0 & y_0 \end{bmatrix}^T\,, \tag{17}$$

$$\boldsymbol{p}_1 = \frac{v_{\lambda 0}}{b}\begin{bmatrix} \cos\varphi_0 & \sin\varphi_0 \end{bmatrix}^T + \boldsymbol{p}_0\,, \tag{18}$$

$$\boldsymbol{p}_2 = \frac{1}{b(b-1)}\begin{bmatrix} \cos\varphi_0 & -v_{\lambda 0}\sin\varphi_0 \\ \sin\varphi_0 & v_{\lambda 0}\cos\varphi_0 \end{bmatrix}\begin{bmatrix} a_{T\lambda 0} \\ \omega_{\lambda 0} \end{bmatrix} + 2\boldsymbol{p}_1 - \boldsymbol{p}_0\,. \tag{19}$$

The influence of these control points is also given graphically in Figure 2.

### 3.2 Combining multiple Bernstein-Bézier curves into a spline

High-order Bernstein-Bézier curves are known to have poor numerical stability and are also computationally expensive to evaluate. Since combining multiple Bernstein-Bézier curves into a single spline enables approximation of an arbitrary path, it is normally recommended to create a path as a spline that consist of multiple low-order Bernstein-Bézier curves — this is always possible. The required smoothness (i.e. order of continuity) of the spline defines the minimum order of the Bernstein-Bézier curves in the spline. A spline that consists of multiple Bernstein-Bézier curves also has a nice property of locality, i.e. each local curve in the spline can be modified to some degree without any effect on the rest of the spline. This can be achieved by moving the *free* control points (i.e. the control points that do not change the desired smoothness properties in the curve boundary points) — if the curve has no such points, order of the curve can always be increased to get this additional degree of freedom. Moreover, a spline can easily be extended without changing the initial curve. Therefore merging of multiple Bernstein-Bézier curves into a spline enables great flexibility when it comes to designing and optimizing a path. Next the essentials of merging two Bézier curves of different orders are presented for the sake of completeness.

We assume that the normalized time $\lambda$ of the spline that consists of $\Lambda \in \mathbb{N}$ Bézier curves runs from 0 to $\Lambda$. This can be achieved with a simple time shifting, i.e. the normalized time of the $i$-th Bézier curve ($i = 1, 2, \ldots, \Lambda$) that runs from 0 to 1 needs to be time shifted into normalized time of the spline from $(i-1)$ to $i$. We will consider a merging of two Bézier curves of different orders, $\boldsymbol{r}_{\lambda,j}(\lambda)$ of order $b_j$ and $\boldsymbol{r}_{\lambda,j+1}(\lambda)$ of order $b_{j+1}$, into a spline. The conditions for merging the curves into a spline can be derived based on the required spline smoothness: continuous spline (continuity

$C^0$), continuous spline with continuous first derivatives (continuity $C^1$), continuous spline with continuous first and second derivatives (continuity $C^2$) or some others.

Continuity $C^0$ of the spline is achieved when the first control point of the Bé curve $j+1$ coincides with the last control of the curve $j$:

$$\boldsymbol{p}_{0,j+1} = \boldsymbol{p}_{b_j,j} \,. \tag{20}$$

To achieve continuity of the angle $\varphi_\lambda$ and velocity $v_\lambda$ also in the spline junctions, the additional conditions for continuity $C^1$ need to be imposed, which yield:

$$\boldsymbol{p}_{1,j+1} = \left(1 + \frac{b_j}{b_{j+1}}\right)\boldsymbol{p}_{b_j,j} - \frac{b_j}{b_{j+1}}\boldsymbol{p}_{b_j-1,j} \,. \tag{21}$$

However, there are some cases that require additional attention. The solution (21) is valid only if $v_{\lambda b_j,j} = v_{\lambda 0,j+1} \neq 0$. In the case $v_{\lambda b_j,j} = v_{\lambda 0,j+1} = 0 \wedge a_{T\lambda b_j,j} = a_{T\lambda 0,j+1} \neq 0$ the following placement of control points achieves continuity $C^1$:

$$\boldsymbol{p}_{2,j+1} = \left(1 + \frac{b_j}{b_{j+1}}\right)\boldsymbol{p}_{b_j,j} - \frac{b_j}{b_{j+1}}\boldsymbol{p}_{b_j-2,j} \,. \tag{22}$$

The required placement of control points for other special conditions (e.g. $v_{\lambda b_j,j} = v_{\lambda 0,j+1} = 0 \wedge a_{T\lambda b_j,j} = a_{T\lambda 0,j+1} = 0 \wedge \dot{a}_{T\lambda b_j,j} = \dot{a}_{T\lambda 0,j+1} \neq 0$) can also be derived. The continuity of spline curvature $\kappa$ is obtained if the spline has continuity $C^2$ that is achieved with the following placement of the control points around the curve junction:

$$\begin{aligned}\boldsymbol{p}_{2,j+1} = &\left(1 + \frac{b_j}{b_{j+1}}\left(2 + \frac{1+b_j}{1+b_{j+1}}\right)\right)\boldsymbol{p}_{b_j,j} - \\ &- 2\left(1 + \frac{b_j(1+b_j)}{b_{j+1}(1+b_{j+1})}\right)\boldsymbol{p}_{b_j-1,j} + \frac{b_j(1+b_j)}{b_{j+1}(1+b_{j+1})}\boldsymbol{p}_{b_j-2,j} \,.\end{aligned} \tag{23}$$

As in the case of (21), the (23) is only valid for $v_{\lambda b_j,j} = v_{\lambda 0,j+1} \neq 0$, otherwise a special solution for the singular case need to be derived.

In the case of non-zero tangential velocity in the curve junction point ($v_{\lambda b_j,j} \neq 0$), the relations (20), (21) and (23) ensure continuity of the curve, curve direction, tangential velocity, angular velocity, curvature and tangential acceleration. Without loss of generality, we therefore require that control points are placed in a way that non-zero tangential velocity is achieved, i.e. all the first and last few control points should not coincide, respectively.

### 3.3 Velocity and turning points

The curvature $\kappa_\lambda(\lambda)$ is given as the fractional-rational function defined in (9). This function has no poles when the velocity $v_\lambda(\lambda)$ is different from zero, but it can have zeros, which then results in the zero curvature. The zeros are therefore the first set of very important points defined by the curvature $\kappa_\lambda(\lambda)$. This set is denoted to belong to a set of velocity points $\mathcal{VP}$. Around these points the velocity can be locally maximal because of small radial acceleration. The second set consists of points where the magnitude of curvature is locally maximal, and it is denoted as the set of turning points $TP$. Both sets together form the set of the motion singularity points denoted as $\mathcal{SP}$.

The derivation of the curvature signum gives the information about the change of the sign in curvature, which are the elements of the set of velocity points $\mathcal{VP} = \{\lambda_{VP_i}\}_{i=1,\ldots,N_{VP}}$, which has $N_{VP}$ elements:

$$\mathcal{VP} \supseteq \mathcal{SP}_1 = \left\{\lambda_i \,:\, \frac{\mathrm{d}\,\mathrm{sign}(\kappa_\lambda(\lambda))}{\mathrm{d}\lambda}\bigg|_{\lambda=\lambda_i} \neq 0\right\} \,. \tag{24}$$

The signum function $\operatorname{sign}(x)$ is defined as:

$$\operatorname{sign}(x) = \begin{cases} 1\,, & \text{if } x \geq 0\,; \\ -1\,, & \text{otherwise.} \end{cases} \tag{25}$$

This is a piecewise-continuous function that has discontinuities where the change of function occurs. The discontinuity points of the signum function $\operatorname{sign}(f(\lambda))$ are identical to the singular points of the signum function derivative with respect to $\lambda$, which can be strictly defined as:

$$\frac{\mathrm{d}\operatorname{sign}(f(\lambda))}{\mathrm{d}\lambda} = \lim_{\epsilon\to 0} \frac{\lim_{\zeta\to\lambda+\epsilon}\operatorname{sign}(f(\zeta)) - \lim_{\zeta\to\lambda-\epsilon}\operatorname{sign}(f(\zeta))}{2\epsilon}\,. \tag{26}$$

Due to the assumption that $v_\lambda(\lambda) > 0$, it holds that $\operatorname{sign}(\kappa_\lambda(\lambda)) = \operatorname{sign}\left(\kappa_\lambda(\lambda)v_\lambda^l(\lambda)\right)$, where $l \in \mathbb{N}$. The set of singularity points $\mathcal{SP}_1$ defined in (24) can therefore also be obtained from

$$\mathcal{VP} \supseteq \mathcal{SP}_1 = \left\{ \lambda_i \,:\, \left.\frac{\mathrm{d}\operatorname{sign}\left(\kappa_\lambda(\lambda)v_\lambda^3(\lambda)\right)}{\mathrm{d}\lambda}\right|_{\lambda=\lambda_i} \neq 0 \right\}. \tag{27}$$

Although this may seem like a more complicated solution, it is normally simpler to evaluate once the term $\kappa_\lambda(\lambda)v_\lambda^3(\lambda)$ is simplified, taking into account (9) and (6), to only

$$\kappa_\lambda(\lambda)v_\lambda^3(\lambda) = x'_\lambda(\lambda)y''_\lambda(\lambda) - x''_\lambda(\lambda)y'_\lambda(\lambda)\,. \tag{28}$$

An example of determining the set of special points $\mathcal{SP}_1 \subseteq \mathcal{VP}$ from the curve curvature is depicted graphically in the top row in Figure 3.

However the set of singularity points $\mathcal{SP}_1$ does not give the complete set of velocity points $\mathcal{VP}$. The remaining velocity points can be found in the set of singularity points $\mathcal{SP}_2$ that is defined as

$$\mathcal{SP}_2 = \left\{ \lambda_i \,:\, \left.\frac{\mathrm{d}\operatorname{sign}\left(\frac{\mathrm{d}\kappa_\lambda(\lambda)}{\mathrm{d}\lambda}\right)}{\mathrm{d}\lambda}\right|_{\lambda=\lambda_i} \neq 0 \right\}. \tag{29}$$

These singularity points are shown in the middle row in Figure 3. Beside the velocity points, the set $\mathcal{SP}_2$ also contains the set of turning points $\mathcal{TP} = \{\lambda_{TP_i}\}_{i=1,\ldots,N_{TP}}$, which has $N_{TP}$ elements and defines the values of the normalized time $\lambda$ where the radius of the curve is locally minimal. An element $\lambda_i \in \mathcal{SP}_2$ belongs to the set of turning points $\mathcal{TP} \subseteq \mathcal{SP}_2$ if the following condition is satisfied:

$$\operatorname{sign}\left(\frac{\mathrm{d}^2\kappa_\lambda(\lambda)}{\mathrm{d}\lambda^2}\kappa_\lambda(\lambda)\right)\bigg|_{\lambda=\lambda_i} < 0 \quad \Rightarrow \quad \lambda_i \in \mathcal{TP} \tag{30}$$

and to the set of velocity points $\mathcal{VP} = \mathcal{SP}_1 \cup (\mathcal{SP}_2 \setminus \mathcal{TP})$ if the next condition is true:

$$\operatorname{sign}\left(\frac{\mathrm{d}^2\kappa_\lambda(\lambda)}{\mathrm{d}\lambda^2}\kappa_\lambda(\lambda)\right)\bigg|_{\lambda=\lambda_i} > 0 \quad \Rightarrow \quad \lambda_i \in \mathcal{VP}\,, \tag{31}$$

which can be observed graphically in the bottom row in Figure 3.

The elements of the set of singularity points $\mathcal{SP} \supseteq \mathcal{SP}_1 \cup \mathcal{SP}_2 \cup \{0, \Lambda\} = \{\lambda_{SP_i}\}$ are arranged in increasing order $\lambda_{SP_i} < \lambda_{SP_{i+1}}$, $i = 1, \ldots, N_{SP} - 1$ where $N_{SP}$ stands for the number of the singularity points. Note that also the curve end points ($\lambda = 0$ and $\lambda = \Lambda$) have been added to the singularity points. If the curvature is a continuous function, then it can be observed that elements in the ordered set of singularity points $\mathcal{SP} = \mathcal{VP} \cup \mathcal{TP}$ ($\mathcal{VP} \cap \mathcal{TP} = \emptyset$) alternate between the elements that belong to the set of velocity points $\mathcal{VP}$ and the elements that belong to the set of turning points $\mathcal{TP}$. Once all the singularity points are found and ordered it therefore suffices to determine only the type of a single element in the set $\mathcal{SP}$ to obtain the sets $\mathcal{VP}$ and $\mathcal{TP}$. Moreover, in the optimization approach presented in Section 3.4 we require that the ordered set

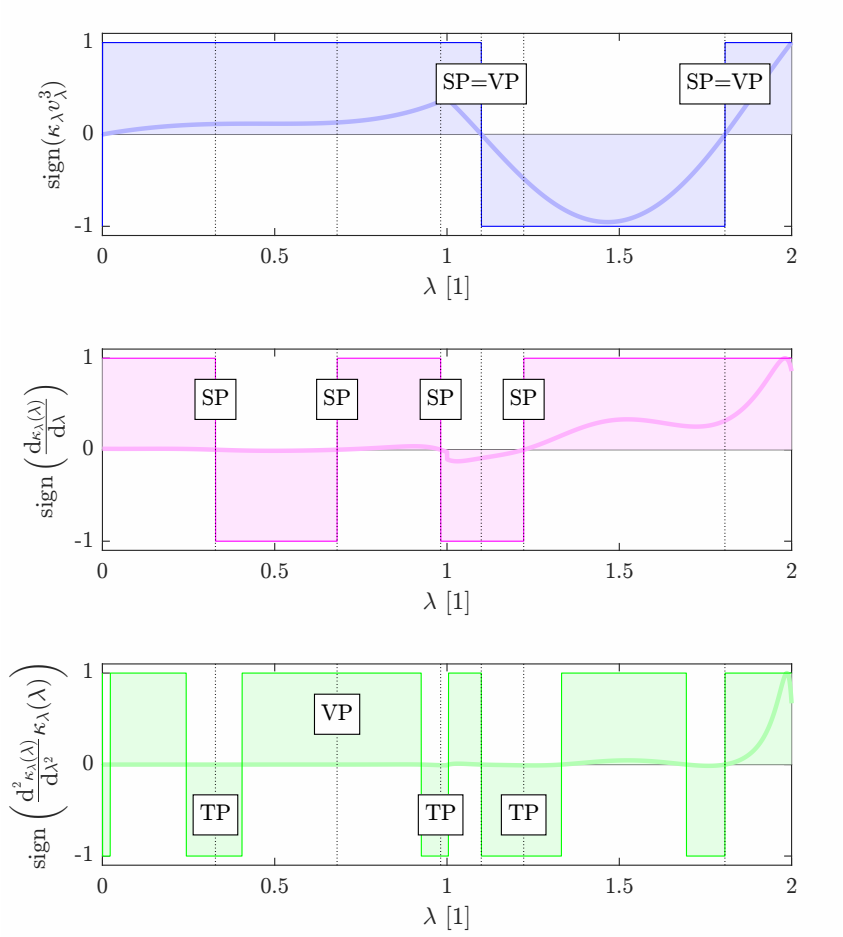


Fig. 3. Determination of singularity (SP), velocity (VP) and turning (TP) points from the continuous curvature of a spline that consist of two Bézier curves.

of singularity points $\mathcal{SP}$ alternates between the velocity and turning points. When combining multiple smooth curves into a spline we therefore enforce that curve merging is made in a way that ensures continuity of curvature over the entire spline as it is presented in Section 3.2.

### 3.4 Velocity-profile optimization

In a turning point the curve curvature is locally maximal. Since the optimization goal is to find the velocity profile that ensures the shortest path-traversal time, the maximum velocity in the turning point that does not violate the constraints need to be determined. In the turning point the tangential acceleration is changing the sign, from negative to positive. So, in the turning point, denoted as $TP_i$, $i = 1, \ldots, N_{TP}$, the tangential acceleration is equal to zero, and it is obtained at normalized time $\lambda_{TP_i}$ and where the curvature equals $\kappa_\lambda(\lambda_{TP_i})$. According to this, the maximum

velocity in the turning point $v_{TP_i}$ due to the allowed radial acceleration $a_{R,max}$ is given as

$$v_{TP_i} = \sqrt{\frac{a_{R,max}}{|\kappa_\lambda(\lambda_{TP_i})|}}\,. \tag{32}$$

Clearly, the angular velocity in the turning point $\omega_{TP_i}$ is defined by (9). Taking into account the velocity constraint $v_{max}$ and angular velocity constraint $\omega_{max}$ the following set of rules apply:

$$\begin{aligned} \text{if} \quad v_{TP_i} > v_{max}\,, \quad \text{then} \quad & v_{TP_i} = v_{max}\,, \\ & \omega_{TP_i} = \kappa_\lambda(\lambda_{TP_i}) v_{TP_i} \end{aligned} \tag{33}$$

and

$$\begin{aligned} \text{if} \quad |\kappa_\lambda(\lambda_{TP_i})| v_{TP_i} > \omega_{max}\,, \quad \text{then} \quad & v_{TP_i} = \frac{\omega_{max}}{|\kappa_\lambda(\lambda_{TP_i})|}\,, \\ & \omega_{TP_i} = \omega_{max} \operatorname{sign}(\kappa_\lambda(\lambda_{TP_i}))\,. \end{aligned} \tag{34}$$

After the constraints are applied, the allowable radial accelerations in the turning points are:

$$a_R(t(\lambda_{TP_i})) = \kappa_\lambda(\lambda_{TP_i}) v_{TP_i}^2\,, \quad i = 1, \ldots, N_{TP}\,. \tag{35}$$

Given a set of $N_{SP}$ ordered singularity points $\mathcal{SP}$ that alternates between velocity and turning points, the segments between the singularity points can be defined as:

$$\mathcal{S}_i = \{\lambda : \lambda_{SP_i} < \lambda < \lambda_{SP_{i+1}}, i = 1, \ldots, N_{SP} - 1\}\,, \tag{36}$$

Around a turning point a locally time-optimal velocity profile can be obtained if we apply maximum allowable acceleration towards the next velocity point and if the turning point is reached with maximum allowable deceleration. To each turning point $TP_j$, $j = 1, \ldots, N_{TP}$ one locally optimum velocity profile is assigned. If $\lambda_{TP_j} = \lambda_{SP_i} \in \mathcal{TP}$, then $\{\lambda_{SP_{i-1}}, \lambda_{SP_{i+1}}\} \in \mathcal{VP}$ and intervals before and after the turning point $\lambda_{TP_j}$ are defined with segments $\mathcal{S}_{i-1}$ and $\mathcal{S}_i$, respectively. The maximum allowable tangential acceleration and deceleration are calculated as proposed in [32]:

$$\begin{aligned} \bar{a}_T(t(\lambda)) &= a_{T,max} \sqrt{1 - \left(\frac{a_R(t(\lambda))}{a_{R,max}}\right)^2}\,, \\ \underline{a}_T(t(\lambda)) &= -a_{T,max} \sqrt{1 - \left(\frac{a_R(t(\lambda))}{a_{R,max}}\right)^2}\,. \end{aligned} \tag{37}$$

Hence, the maximum allowable increase of the tangential velocity due to the acceleration $\bar{a}_T(\lambda)$ along the interval $\mathcal{S}_i$ ($\lambda_{TP_j} = \lambda_{SP_i} < \lambda < \lambda_{SP_{i+1}}$) is:

$$v(t(\lambda)) - v_{TP_j} = \int_{t(\lambda_{TP_j})}^{t(\lambda)} \bar{a}_T(\tau)\,\mathrm{d}\tau = \int_{\lambda_{TP_j}}^{\lambda} \bar{a}_T(t(\zeta)) \frac{v_\lambda(\zeta)}{v(t(\zeta))}\,\mathrm{d}\zeta\,. \tag{38}$$

Similarly, the maximum allowable tangential deceleration is enforced in the interval $S_{i-1}$ ($\lambda_{SP_{i-1}} < \lambda < \lambda_{SP_i} = \lambda_{TP_j}$) that ensures the desired tangential velocity in the turning point $TP_j$:

$$v_{TP_j} - v(t(\lambda)) = \int_{t(\lambda_{j,TP})}^{t(\lambda)} \underline{a}_T(\tau)\,\mathrm{d}\tau = \int_{\lambda_{j,TP}}^{\lambda} \underline{a}_T(t(\zeta)) \frac{v_\lambda(\zeta)}{v(t(\zeta))}\,\mathrm{d}\zeta\,. \tag{39}$$

In integrations (38) and (39) the constraints need to be taken into account:

$$\text{if} \quad \left|\frac{a_R(t(\lambda))}{\kappa_\lambda(\lambda)}\right|^{\frac{1}{2}} > v_{max}\,, \quad \text{then} \quad \begin{aligned} a_R(t(\lambda)) &= \kappa_\lambda(\lambda) v_{max}^2\,, \\ v(t(\lambda)) &= v_{max}\,, \\ \omega(t(\lambda)) &= \kappa_\lambda(\lambda) v_{max} \end{aligned} \tag{40}$$

and

$$\text{if} \quad |a_R(t(\lambda))\kappa_\lambda(\lambda)|^{\frac{1}{2}} > \omega_{max}\,, \quad \text{then} \quad \begin{aligned} a_R(t(\lambda)) &= \frac{\omega_{max}^2}{|\kappa_\lambda(\lambda)|}\,, \\ v(t(\lambda)) &= \frac{\omega_{max}}{\kappa_\lambda(\lambda)}\,, \\ \omega(t(\lambda)) &= \omega_{max}\text{sign}(\kappa_\lambda(\lambda))\,. \end{aligned} \tag{41}$$

Each local optimum velocity profile needs to be determined on a domain $\mathcal{L}_j$ that consist of a deceleration domain $\mathcal{D}_j$ and an acceleration domain $\mathcal{A}_j$, i.e. $\mathcal{L}_j = \mathcal{A}_j \cup \mathcal{D}_j$, which do not overlap ($\mathcal{A}_j \cap \mathcal{D}_j = \emptyset$). The acceleration domain always consists of segment $\mathcal{S}_i$, i.e. $\mathcal{S}_i \subseteq \mathcal{A}_j$, and deceleration domain always contains segment $\mathcal{S}_{i-1}$, i.e. $\mathcal{S}_{i-1} \subseteq \mathcal{D}_j$. However in order to determine the global optimum velocity profile by merging the local optimum profiles, the integration bounds in (38) and (39) might need to be extended to the neighboring segments. The acceleration and deceleration segments are therefore:

$$\begin{aligned} \mathcal{A}_j &= \mathcal{S}_i \cup \mathcal{S}_{i+1} \cup \ldots \cup \mathcal{S}_{i+a_j-1}\,, \\ \mathcal{D}_j &= \mathcal{S}_{i-1} \cup \mathcal{S}_{i-2} \cup \ldots \cup \mathcal{S}_{i-d_j}\,, \end{aligned} \tag{42}$$

where $a_j, d_j \in \mathbb{N}$, $1 \leq a_j \leq N_{SP} - i$ and $1 \leq d_j \leq i-1$. Although the locally optimum velocity profile could be evaluated over all available segments, this is normally not required and can consume a lot of additional computational time, especially if the path is very long (i.e. when the spline is comprised of many curves).

Figures 4 and 5 show an example of local optimum tangential and angular velocity profiles for every domain around each of the turning points, respectively. The grayed-out area in the figures represent the inadmissible region that is defined by the velocity constraints given in (32) to (34). It can be observed that the domains of local velocity profiles overlap. The size of overlapping is determined in a way that the global velocity profile can be obtained from the optimum local velocity profiles and that each local velocity profile can still be evaluated independently of each other, as it is presented next.

First, we present the conditions that enable determination of sufficient range of acceleration and deceleration domains that are defined with the integers $a_j$ and $d_j$. The conditions are not also necessary, but have been defined in a way that enable independent calculation of each optimum local velocity profile. Notice that all the segments $\mathcal{S}_{i+2p}$ ($p \in \mathbb{Z}$) start in a turning point and end in a velocity point and all the segments $\mathcal{S}_{i+2p-1}$ ($p \in \mathbb{Z}$) start in a velocity point and end in a turning point. Let us assume that the locally optimum velocity profile that is defined on domain $\mathcal{L}_j$ is denoted as $v_{\mathcal{L}_j}(t(\lambda))$, $j = 1, 2, \ldots, N_{TP}$. After the integral (38) is evaluated over each consequent domain in $\mathcal{A}_j$ the following conditions are evaluated. If the integration ended in a velocity point and the integrated velocity $v_{\mathcal{L}_j}$ in that point is less than the velocity constraint $v_{max}$, then the integration should continue to the next segment — e.g. this can be observed in Figure 4 on the acceleration domain $\mathcal{A}_2 \subseteq \mathcal{L}_2$. Else if the integration ended in a turning point and the integrated velocity in that point is less that the velocity in the corresponding velocity point, then the integration should also continue to the next segment. Otherwise, the integration over the remaining segments is not

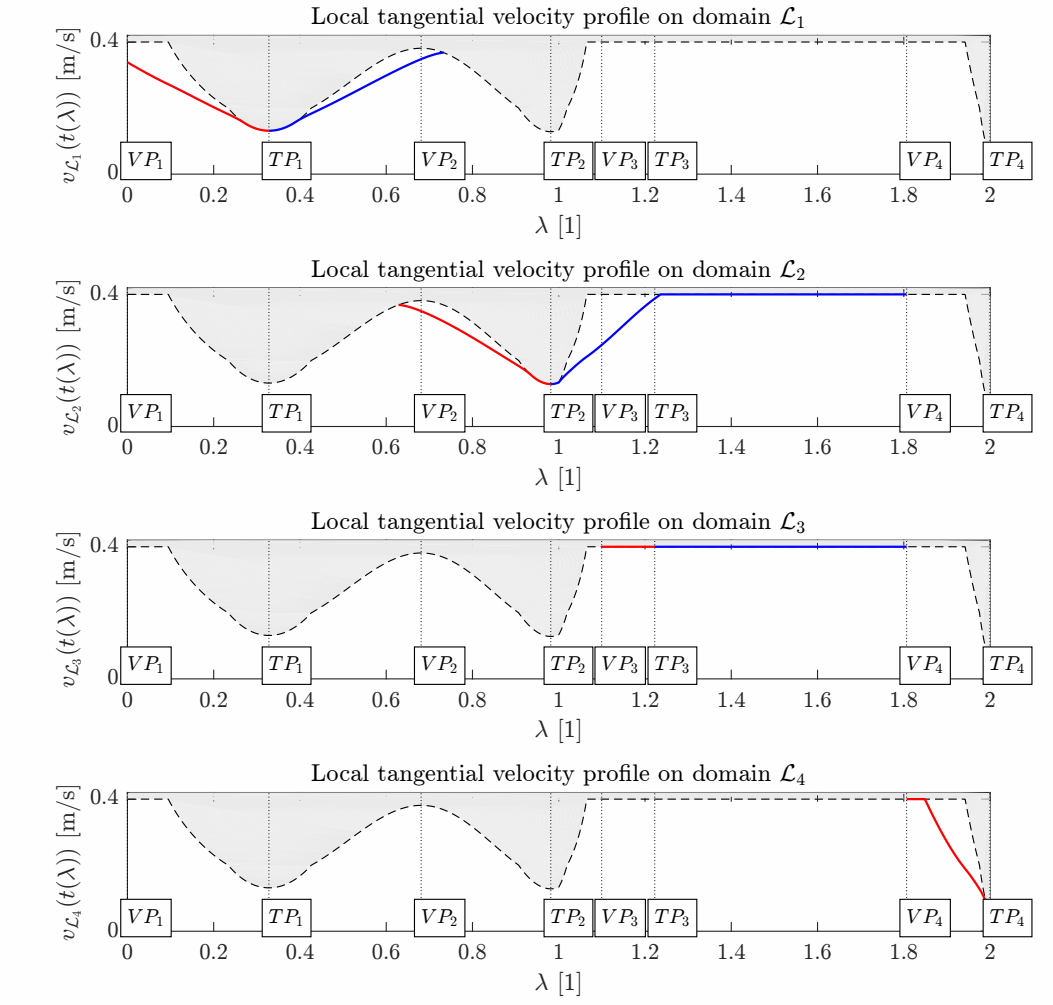


Fig. 4. Local optimum tangential velocity profiles on the local domains around all the turning points. The grayed-out area bounded by the dashed line represents the inadmissible region where the velocity constrains cannot be respected.

required, i.e. the last segment over which the integration has been made is $\mathcal{S}_{i+a_j-1}$ — e.g. in Figure 4 the acceleration domain $\mathcal{A}_2 \subseteq \mathcal{L}_2$ and the acceleration domain $\mathcal{A}_3 \subseteq \mathcal{L}_3$ both end at the velocity point $VP_4$. When the integration is made from a velocity point towards the turning, the integration can be terminated as soon as the boundary of inadmissible velocity region (determined in (32) to (34)) is reached — e.g. this can be observed in Figure 4 on the acceleration domain $\mathcal{A}_1 \subseteq \mathcal{L}_1$. The same conditions apply when determining the range of deceleration segments $\mathcal{D}_j$ and the number $d_j$. Local velocity profiles are optimized on a domain larger than the minimum necessary domain for determination of the global optimum velocity profile. However, there are some other benefits. Since each local velocity profile can be optimized independently of the other local velocity profiles, this part of the optimization algorithm is suitable for concurrent implementation. Furthermore, if

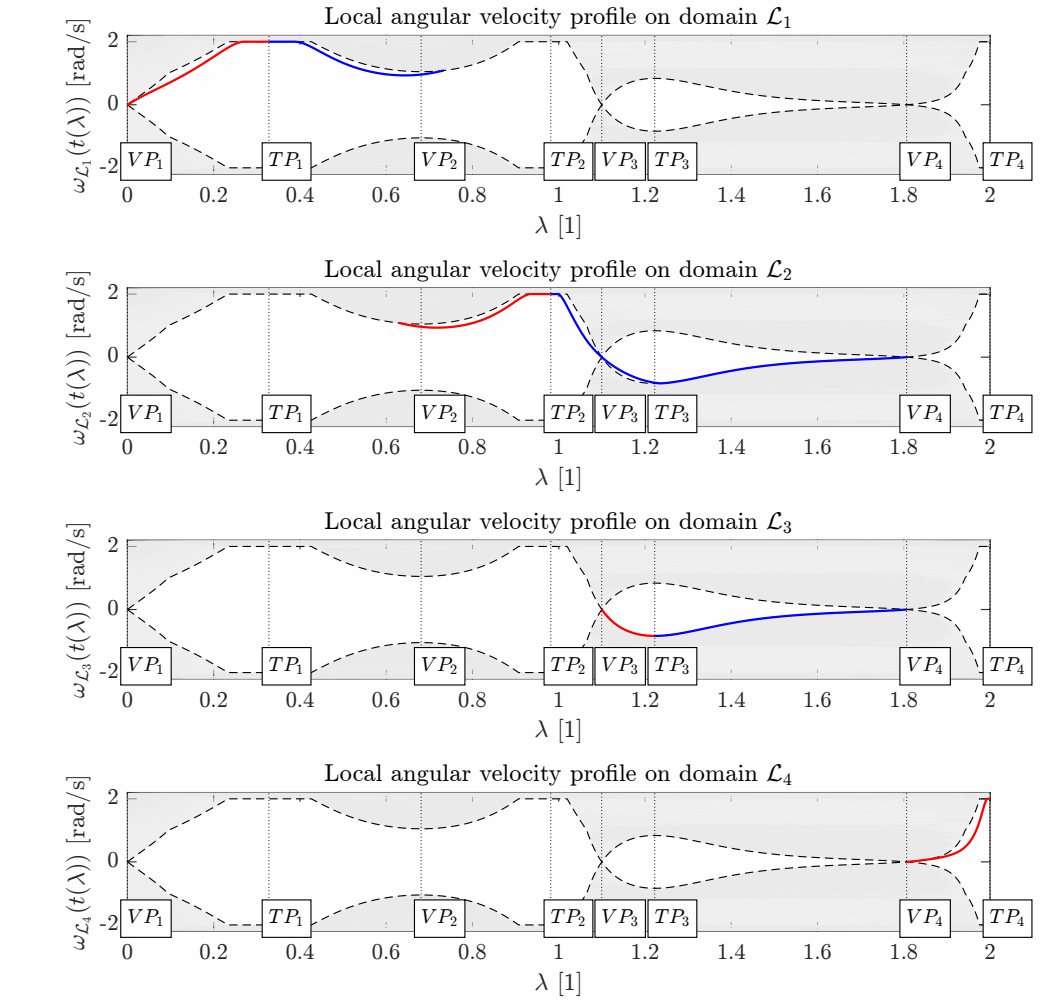


Fig. 5. Local optimum angular velocity profiles on the local domains around all the turning points. The grayed-out area bounded by the dashed line represents the inadmissible region where the velocity constrains cannot be respected.

the path is extended or modified only the local velocity profiles on the modified segments need to be (re)calculated.

Let us make a remark on optimization at the segments in the ends of the path. If the path ends in a velocity point, the optimization of local profiles is as described above. However, if the end point is a turning point, then the local velocity profile need to be determined only on acceleration or deceleration segments, whether the turning point is the curve end or start point, respectively. If particular initial and final velocities are also required, it is straight forward to incorporate these if the end points are turning points. In the case the end point is a velocity point, additional local velocity profile need to be introduced and the same optimization approach need to be taken in a way that the end point is considered to behave like a turning point.

Once all the locally optimum velocity profiles are obtained the globally optimum velocity profile can be determined. The overall velocity $v(\lambda)$ is obtained if all the segments $v_{\mathcal{L}_j}(t(\lambda))$ ($\lambda \in \mathcal{L}_j$) are merged together together as follows:

$$v(t(\lambda)) = \min_j v_{\mathcal{L}_j}(t(\lambda)) \,, \tag{43}$$

where the minimization is made only over the velocity profiles $v_{\mathcal{L}_j}(t(\lambda))$ that contain the normalized time $\lambda$ in their domain of definition, i.e. $\lambda \in \mathcal{L}_j$.

Finally, the complete time function $t(\lambda)$ that gives the relation between the normalized and real time can be obtained from integration of (3):

$$t(\lambda) = \int_0^{t(\lambda)} \mathrm{d}\tau = \int_0^{\lambda} \frac{v_\lambda(\zeta)}{v(t(\zeta))}\,\mathrm{d}\zeta \tag{44}$$

The total time $T_{fin}$ that is needed to traverse the entire path is therefore:

$$T_{fin} = t(\Lambda) - t(0) = \int_0^{\Lambda} \frac{v_\lambda(\zeta)}{v(\zeta)}\,\mathrm{d}\zeta \,. \tag{45}$$

Let us present a proof that this global velocity profile that has been obtained from local optimum velocity profiles is indeed an optimum solution. First we can observe that the ordered set of singularity points alternates between the velocity and turning points. Since each local optimum velocity profile is evaluated at least in the domain between the two adjacent velocity points of the turning point for all the turning points, the whole domain of the path is covered by the local velocity profiles. We can also observe that the lower boundary of admissible velocity region that is defined by the equations (32) to (34) is continuous (the dashed lines in Figures 4 and 5). The end points of each optimum local velocity profile coincide with the aforementioned boundary of admissible velocity region. Therefore the adjacent local optimum velocity profiles must be intersecting and that means that the optimum velocity profile calculated from (43) is continuous. If the global velocity profile would be calculated from local optimum velocity profiles in any other way, this would introduce discontinuities, which are inevitably connected with violation of acceleration constraints. However, there is an infinite family of continuous velocity profiles $v_\star(t)$ that have tangential velocity below the global tangential velocity calculated from (43), i.e. $v_\star(t) \leq v(t)$. Since the length of the path associated with each of the velocity profile need to be the same:

$$\int_0^{T_{fin}} v(t)\,\mathrm{d}t = \int_0^{T_{fin,\star}} v_\star(t)\,\mathrm{d}t \,, \tag{46}$$

it is obvious that the traversal time of any other velocity profile that the one obtained from (43) is longer, i.e. $T_{fin} \leq T_{fin,\star}$. Therefore the global velocity profile obtained from the local optimum velocity profiles according to (43) is indeed optimal.

## 4 EXPERIMENTS

### 4.1 Simulation

For the purpose of simulation we define a path that consists of two fourth order Bernstein-Bézier curves that are merged into a spline in a way that continuity $C^2$ of the path is achieved. The five control points $P_0$ to $P_4$ define the first part of the path and the five control points $R_0$ to $R_4$ define the second part of the path. The control point $P_0$ defines the desired initial point and the control point $R_4$ the desired end point of the path. The control points $P_1$ and $R_3$ are defined in a way that the

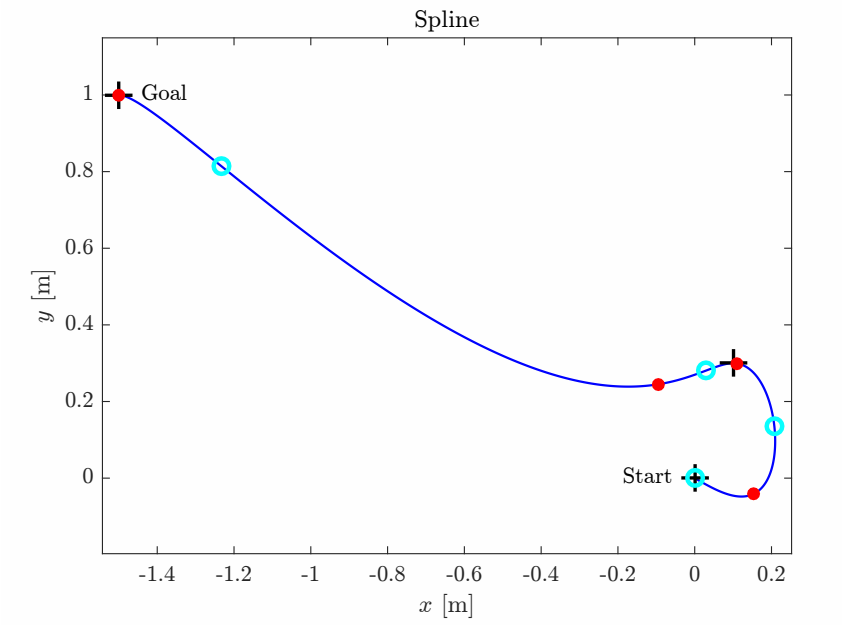


Fig. 6. Spline of two Bézier curves (each curve end points are marked with crosses) with marked velocity points VP (hollow disks) and turning points TP (filled circles).

Table 1. The control points of two Bernstein-Bézier curves that form a smooth spline for the purpose of simulation.

| Curve 1 control points | $P_0$ [m] | $P_1$ [m] | $P_2$ [m] | $P_3$ [m] | $P_4$ [m] |
|---|---|---|---|---|---|
| | $(0, 0)$ | $(0.13, -0.075)$ | $(0.26, -0.15)$ | $(0.25, 0.3)$ | $(0.1, 0.3)$ |
| Curve 2 control points | $R_0$ [m] | $R_1$ [m] | $R_2$ [m] | $R_3$ [m] | $R_4$ [m] |
| | $(0, 0)$ | $(0.13, -0.075)$ | $(0.26, -0.15)$ | $(0.25, 0.3)$ | $(0.1, 0.3)$ |

velocity and orientation requirements at the path end points are satisfied. Normally the placement of initial control points should reflect the initial pose of the mobile robot and the final control points the desired goal pose at the end of the path. The control point $P_4$ defines the desired intermediate point that the path should pass through. The control points $P_2$ and $P_3$ can be placed in a way that additional desired properties at the path start or intermediate point are achieved. To achieve path continuity, continuous tangential velocity and also continuous curvature of the spline, the control points $R_0$, $R_1$ and $R_2$ must be placed with respect to the control points $P_2$, $P_3$ and $P_4$ according to the rules (20), (21) and (23). According to the aforementioned rules we defined all the control points that are presented in Table 1. The path that is defined by these control points is shown in Figure 6. Note that the spline consists of two curves with very different lengths.

The goal of the optimization is to find the optimum velocity profile for the path given in Figure 6 as a spline of Bézier curves in a way that the following set of velocity and acceleration constraints

is satisfies:

$$\begin{aligned} v_{max} &= 0.4\,\mathrm{m/s}\,, \quad \omega_{max} = 2\,\mathrm{rad/s}\,, \\ a_{T,max} &= 0.5\,\mathrm{m/s^2}\,, \quad a_{R,max} = 0.4\,\mathrm{m/s^2}\,. \end{aligned} \tag{47}$$

The most simple way of satisfying these constraints is to use a simple linear time function $t(\lambda) \propto \lambda$, since it's first derivative with respect to $\lambda$ is constant and all higher-order derivatives are zero. The relations between the velocities and accelerations in normalized and real time become trivial. With appropriate scaling of the time function the constraints can be satisfied. Clearly this solution is very conservative.

An optimum velocity profile that is obtained with the proposed approach is presented in Figure 7. No velocity constraints are violated, and also the acceleration constraints are satisfied as it can be observed from Figures 8 and 9. The optimized time function is shown in Figure 10, which is clearly non-linear. The time needed to pass through the first curve is shorter that the time that is required to pass through the second curve, although the normalized times of both curves are equal. Figure 3 shows how the singularity points have been determined for the spline in Figure 6, where the locations of the velocity and the turning points are also marked. It can be seen that turning points and velocity points alternate in the set of singularity points, as it is expected since the curve curvature is a continuous function. The local optimum velocity profiles for the curve in Figure 6 can be seen in Figures 4 and 5. These simulation results confirm applicability of the proposed optimization approach.

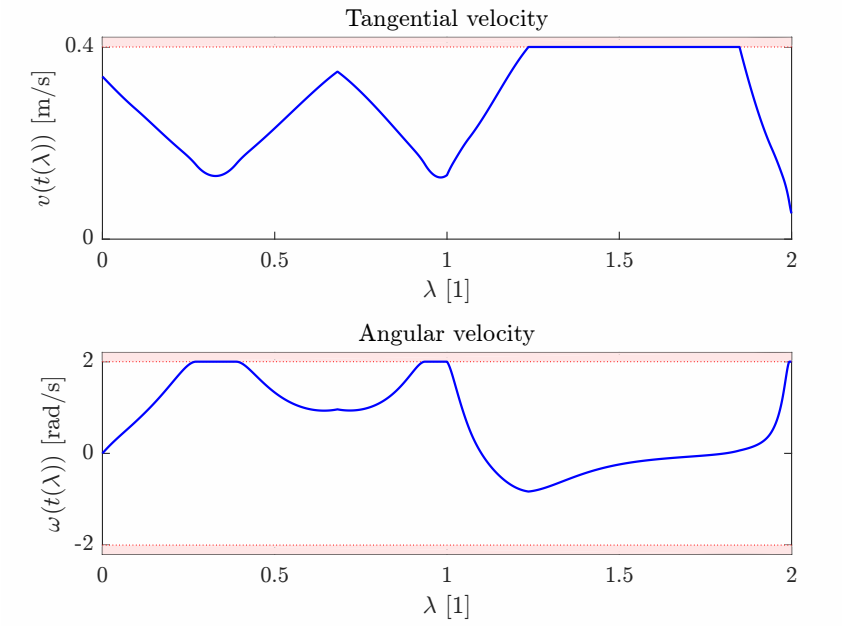


Fig. 7. Optimized velocity profile that satisfies the velocity and acceleration constraints.

### 4.2 Real robots

The usage of the presented velocity profile optimization has been verified on a real mobile robot. For this purpose a small differentially driven wheeled mobile robot that is normally used in Micro Robot

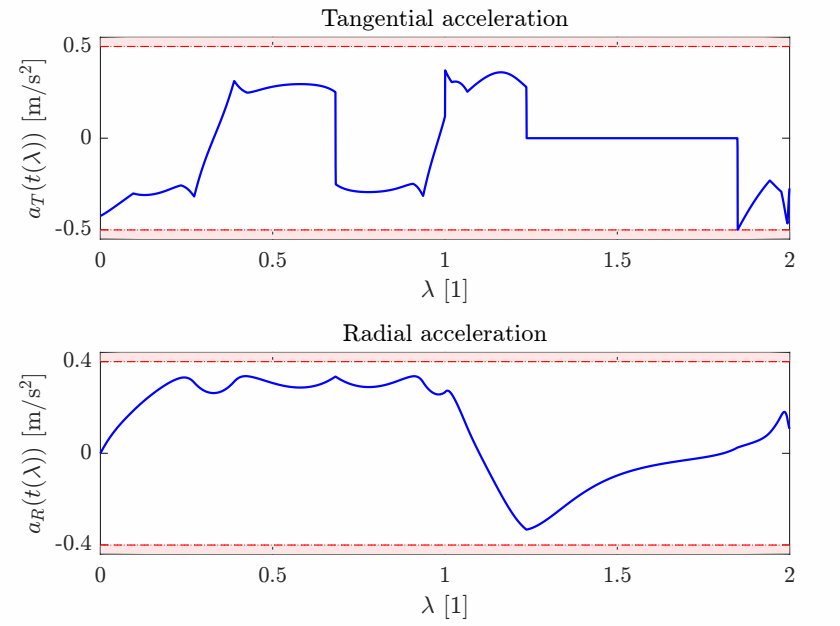


Fig. 8. Acceleration profile that satisfies the velocity and acceleration constraints.

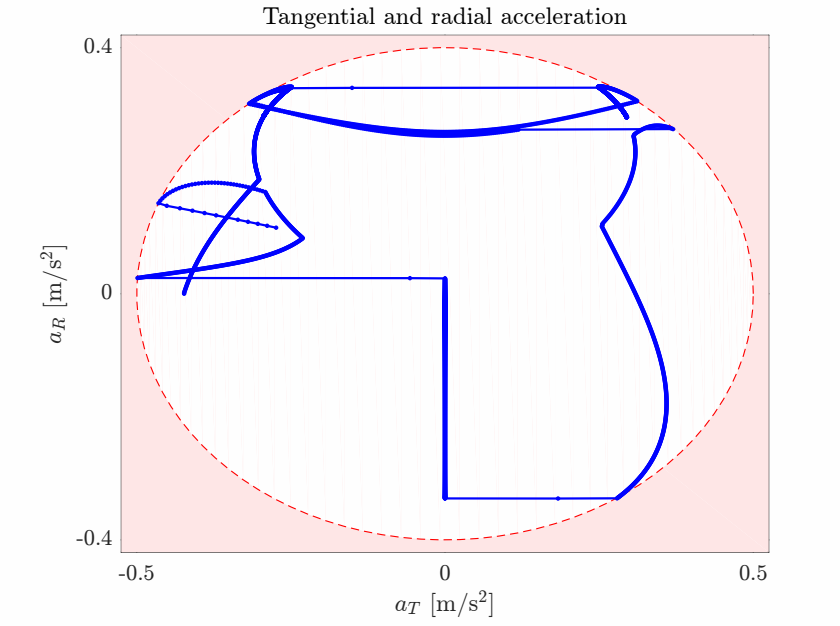


Fig. 9. Diagram of the constrained tangential and radial acceleration.

Soccer Tournaments (MiRoSoT) has been used. The mobile robot has implemented an internal velocity controller in a way that the mobile robot can be controlled by the desired tangential and

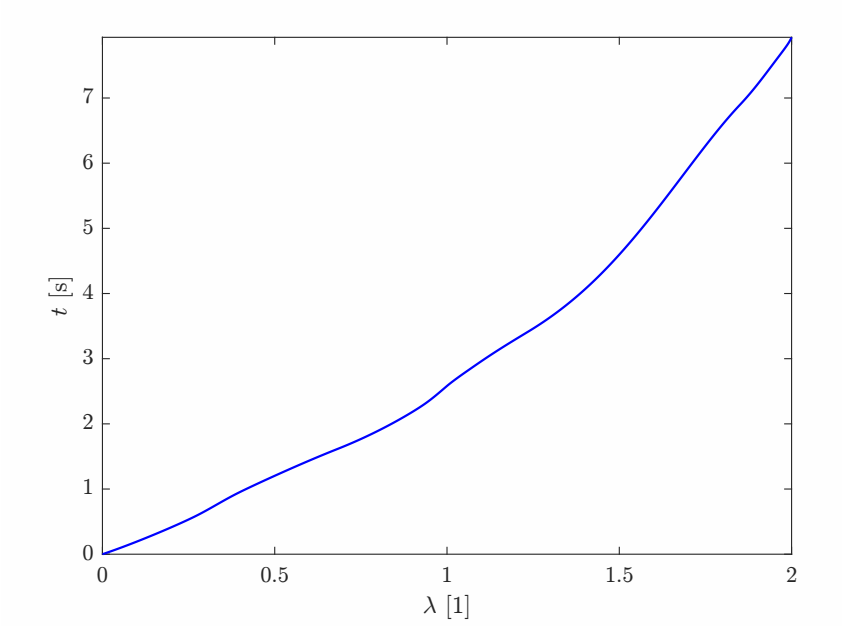


Fig. 10. Optimized time function that achieves the optimum velocity profile under the velocity and acceleration constraints.

angular velocity. On top of the mobile robot is a special marker that can be observed by an overhead camera and tracked by a machine vision system that runs in real-time with a frequency of 25 Hz and enables implementation of mobile robot pose tracking. The vision tracking system can measure position and orientation of the robot in the ground plane. These measurements can be processed in order to estimate robot velocities and accelerations (first and second order differentiation of the measured position). The robot was additionally equipped with a Inertial Measuring Unit (IMU) that consists of a 3-axis gyroscope and accelerometer. Since a low cost IMU has been used, the acceleration measurements contained a lot of noise and an appropriate low-pass filter needed to be used.

For the purpose of the experiment a smooth trajectory that consists of four Bézier curves was generated (Figure 11) in a similar way as presented in the simulation experiment. A two-degrees-of-freedom control approach is normally used in the implementation of a trajectory tracking problem (e.g. the predictive control algorithm [29]). A feed-forward part is used to drive the system into the vicinity of the reference trajectory and a feedback part is used to compensate for the tracking error that may occur due to imprecise system modeling, noise and other disturbances. However, in the experiments only a feed-forward controller was used in order to verify that the proposed optimization approach can generate a velocity profile that does not violate velocity and acceleration constraints and also that the shape of the curve is preserved. Nevertheless, each robot drive wheel has an internal velocity PID controller, therefore the mobile robot can be controlled with the desired forward and angular velocities. Again, the requirements (47) were used in the optimization of the velocity profile. If these requirements are respected, the dynamics of the internal wheel velocity controllers are negligible. The results of the experiment are shown in Figures 11 to 13 where the measured data from the visual tracking system is overlaid over the generated trajectory data. In the bottom of Figure 12 also the measurements from the on-board gyroscope are shown, and in

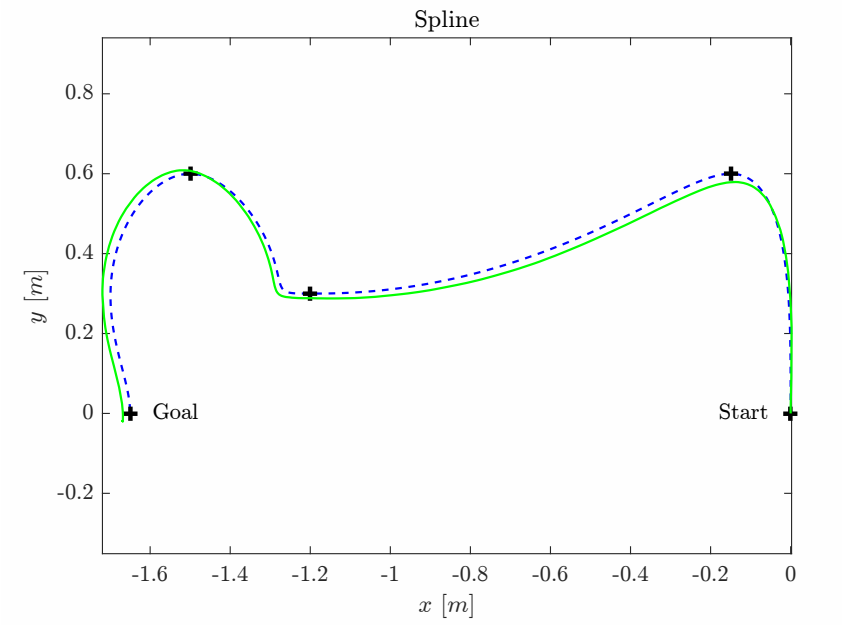


Fig. 11. Reference (dashed line) and measured path (solid line) that consists of four Bernstein-Bézier curves (curve end points are marked with a cross).

Figure 13 measurements from the on-board accelerometer are also given. Although there is some measurement noise present, the results confirm that the trajectory generated with the proposed approach satisfies all the velocity and acceleration constraints and that the output path is close to the desired path. This confirms the applicability of the propose approach.

## 5 CONCLUSION

In this paper an approach for velocity profile optimization of a Bernstein-Bézier spline has been presented. The conditions that enable smooth generation of a spline from multiple curves that may not all be of the same order have been presented. Generation of a spline with a continuous curvature enabled usage of the proposed algorithm for velocity profile optimization. The proposed optimization approach can take into account the velocity and acceleration constraints and produce a continuous velocity profile. The optimum velocity profile has been generated based on the velocity and turning points. These singularity points were used to define the conditions that limit the optimization of local velocity profiles to local vicinity of each turning point, but their range is sufficient for determination of the global optimum. The proposed algorithm exhibits some locality property that in the case the path is modified or extended only the optimum local velocity profiles on the modified or new segments need to be optimized again. Since the local optimum velocity profiles can be evaluated independently of each other, the proposed approach is also suitable for concurrent implementation. Verification of the proposed approach has been made in the simulation environment and on real mobile robot. The approach enables optimization of a spline velocity profile even in the case that the curves forming the spline are of substantially different lengths.

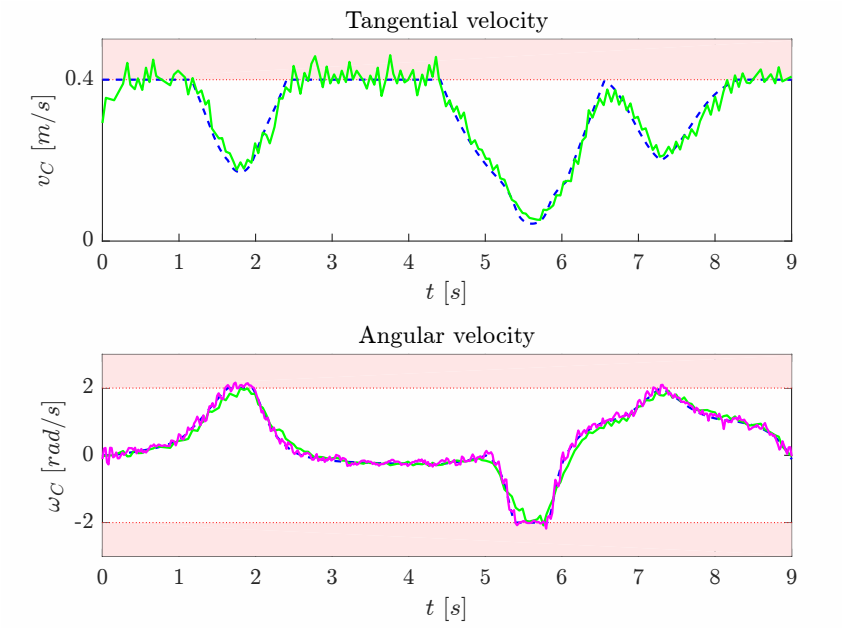


Fig. 12. Velocity profile that satisfies the velocity and acceleration constraints: calculated velocity (dashed line), estimated velocity with the visual measuring system (green solid line) and measured angular velocity with the on-board IMU gyroscope (purple solid line).

## REFERENCES


[1] M. Al-Khatib and Jean J. Saade. 2003. An efficient data-driven fuzzy approach to the motion planning problem of a mobile robot. *Fuzzy Sets and Systems* 134, 1 (2003), 65–82.

[2] Aijun Bai, Feng Wu, and Xiaoping Chen. 2015. Online planning for large markov decision processes with hierarchical decomposition. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 4 (2015), 45.

[3] L. Benamati, C. Cosma, and P. Fiorini. 2005. Path planning using flat potential field approach. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*. IEEE, 103–108.

[4] C. Guarino Lo Bianco. 2013. Minimum-Jerk Velocity Planning for Mobile Robot Applications. *IEEE Transactions on Robotics* 29, 5 (Oct. 2013), 1317–1326.

[5] S. Blažič. 2014. On Periodic Control Laws for Mobile Robots. *IEEE Transactions on Industrial Electronics* 61, 7 (July 2014), 3660–3670.

[6] D. Cagigas. 2005. Hierarchical D★ algorithm with materialization of costs for robot path planning. *Robotics and Automonous Systems* 52 (2005), 190–208.

[7] Debasri Chakraborty, Warren Vaz, and Arup Kr. Nandi. 2015. Optimal driving during electric vehicle acceleration using evolutionary algorithms. *Applied Soft Computing* 34 (2015), 217–235.

[8] W. Chen and W. He. 2009. Bézier curve based path planning for mobile robot. *Robot Technique and Application* 5 (2009), 56–59.

[9] H. Choset and J. Burdick. 2000. Sensor based motion planning: Incremental construction of the hierarchical generalized Voronoi graph. *International Journal of Robotics Research* 19, 2 (2000), 126–148.

[10] Marija Dakulović and Ivan Petrović. 2011. Two-way D* algorithm for path planning and replanning. *Robotics and Autonomous Systems* 59, 5 (2011), 329–342. Special Issue ECMR 2009.

[11] Mansoor Davoodi, Fatemeh Panahi, Ali Mohades, and Seyed Naser Hashemi. 2015. Clear and smooth path planning. *Applied Soft Computing* 32 (2015), 568–579.

[12] G. Desaulniers. 1996. On shortest paths for a car-like robot maneuvering around obstacles. *Robotics and Autonomous Systems* 17 (1996), 139–148.

[13] M. Egerstedt and X. Hu. 2002. A hybrid control approach to action coordination for mobile robots. *Automatica* 38 (2002), 125–130.

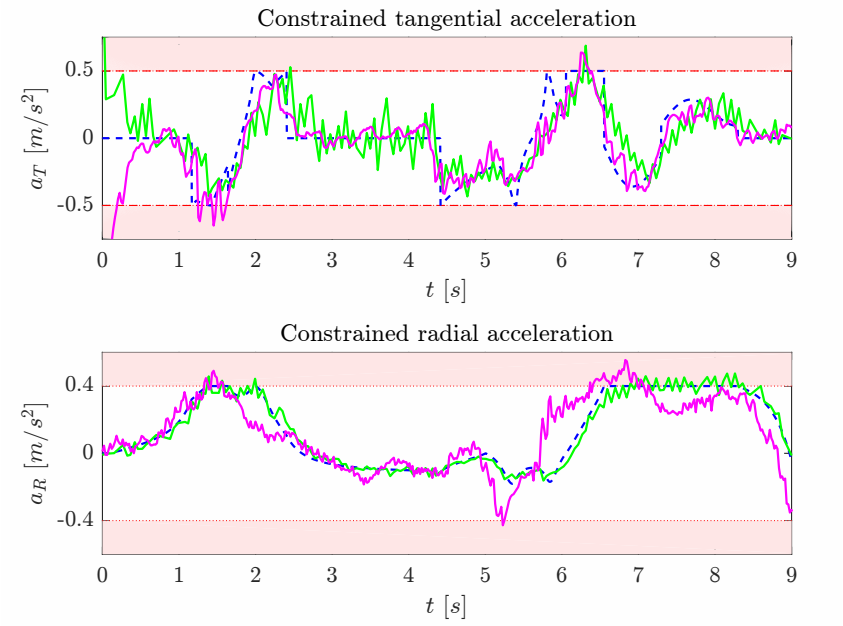


Fig. 13. Acceleration profile that satisfies the velocity and acceleration constraints: calculated acceleration (dashed line), estimated acceleration with the visual measuring system (green solid line) and measured acceleration with the on-board IMU accelerometer sensor (purple solid line).

[14] Tara A. Estlin, Benjamin J. Bornstein, Daniel M. Gaines, Robert C. Anderson, David R. Thompson, Michael Burl, Rebecca Casta no, and Michele Judd. 2012. AEGIS Automated Science Targeting for the MER Opportunity Rover. *ACM Trans. Intell. Syst. Technol.* 3, 3 (May 2012), 50:1–50:19.

[15] D. Ferguson and A. Stentz. 2006. Using interpolation to improve path planning: The field D⋆ algorithm. *Journal of Field Robotics* 23, 2 (2006), 79–101.

[16] A. Fujimori, P. N. Nikiforuk, and M. M. Gupta. 1999. Adaptive navigation of mobile robots with obsticle avoidance. *IEEE Transactions on Robotics and Automation* 13, 4 (1999), 596–602.

[17] A. I. Ginnis and P. D. Kaklis. 2002. Planar C2 cubic spline interpolation under geometric boundary conditions. *Computer Aided Geometric Design* 19, 5 (2002), 345–363.

[18] M. Haddad, W. Khalil, and H. E. Lehtihet. 2010. Trajectory Planning of Unicycle Mobile Robots With a Trapezoidal-Velocity Constraint. *IEEE Transactions on Robotics* 26, 5 (Oct. 2010), 954–962.

[19] Y. J. Ho and J. S. Liu. 2009. Collision-free curvature-bounded smooth path planning using composite Bézier curve based on Voronoi diagram. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*. IEEE, 463–468.

[20] H. P. Huang and C. C. Liang. 2002. Strategy-based decision making of a soccer robot system using a real-time self-organizing fuzzy decision tree. *Fuzzy Sets and Systems* 127, 1 (2002), 49–64.

[21] Kao-Shing Hwang and Ming-Yi Ju. 2002. A propagating interface model strategy for global trajectory planning among moving obstacles. *IEEE Transactions on Industrial Electronics* 49, 6 (Dec. 2002), 1313–1322.

[22] Yong K. Hwang and Narendra Ahuja. 2005. A potential field approach to path planning. *IEEE Transactions on Robotics and Automation* 8, 1 (2005), 23–32.

[23] X. J. Jing. 2005. Behaviour dynamics based motion planning of mobile robots in uncertain dynamic environments. *International Journal of Robotics and Autonomous Systems* 53 (2005), 99–123.

[24] K. G. Jolly, R. Sreerama Kumar, and R. Vijayakumar. 2009. A Bézier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits. *Robotics and Autonomous Systems* 57, 1 (2009), 23–33.

[25] M. Kazemi, M. Mehrandezh, and K. Gupta. 2005. Sensor-based robot path planning using harmonic function-based probabilistic roadmaps. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*. IEEE, 84–89.

[26] K. Kiguchi, K. Watanabe, and T. Fukuda. 2001. Trajectory planning of mobile robots using DNA computing. In *IEEE Int. Conf. on Computational Intelligence in Robotics and Automation*. IEEE, 380–385.
[27] D. H. Kim and J. H. Kim. 2002. A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer. *International Journal of Robotics and Autonomous Systems* 42 (2002), 17–30.
[28] H. Kim and B. K. Kim. 2014. Online Minimum-Energy Trajectory Planning and Control on a Straight-Line Path for Three-Wheeled Omnidirectional Mobile Robots. *IEEE Transactions on Industrial Electronics* 61, 9 (Sept. 2014), 4771–4779.
[29] G. Klančar and I. Škrjanc. 2007. Tracking-error model-based predictive control for mobile robots in real time. *Robotics and Autonomous Systems* 55, 6 (2007), 460–469.
[30] G. Klančar and I. Škrjanc. 2010. A case study of the collision-avoidance problem based on Bernstein-Bézier path tracking for multiple robots with known constraints. *Journal of Intelligent and Robotic Systems* 60, 2 (2010), 317–337.
[31] J. P. Laumond. 1998. *Robot Motion Planning and Control*. Lecture Notes in Control and Information Science, Vol. 229. Springer-Verlag. 347 pages.
[32] M. Lepetič, G. Klančar, I. Škrjanc, D. Matko, and B. Potočnik. 2003. Time optimal planning considering acceleration limits. *Robotics and Autonomous Systems* 45 (2003), 199–210.
[33] T. C. Liang, J. S. Liu, G. T. Hung, and Y. Z. Chang. 2005. Practical and flexible path planning for car-like mobile robot using maximal-curvature cubic spiral. *International Journal of Robotics and Autonomous Systems* 52 (2005), 312–335.
[34] D. Liu, M. Cong, and Y. Du. 2017. Episodic Memory-Based Robotic Planning Under Uncertainty. *IEEE Transactions on Industrial Electronics* 64, 2 (Feb. 2017), 1762–1772.
[35] Thi Thoa Mac, Cosmin Copot, Duc Trung Tran, and Robin De Keyser. 2017. A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. *Applied Soft Computing* 59 (2017), 68–76.
[36] F. M. Marchese. 2002. A directional diffusion algorithm on cellular automata for robot path-planning. *Future Generation Computer Systems* 18 (2002), 983–994.
[37] C. F. Martin, S. Sun, and M. Egerstedt. 2001. Optimal control, statistics and path planning. *Mathematical and Computer Modelling* 33 (2001), 237–253.
[38] E. Ozatay, S. Onori, J. Wollaeger, U. Ozguner, G. Rizzoni, D. Filev, J. Michelini, and S. Di Cairano. 2014. Cloud-Based Velocity Profile Optimization for Everyday Driving: A Dynamic-Programming-Based Solution. *IEEE Transactions on Intelligent Transportation Systems* 15, 6 (Dec. 2014), 2491–2505.
[39] E. Papadopoulos, I. Papadimitriou, and I. Poulakakis. 2005. Polynomial-based obstacle avoidance techniques for nonholonomic mobile manipulator systems. *Robotics and Autonomous System* 51 (2005), 229–247.
[40] Toni Petrinić, Mišel Brezak, and Ivan Petrović. 2017. Time-optimal velocity planning along predefined path for static formations of mobile robots. *International Journal of Control, Automation and Systems* 15, 1 (Feb. 2017), 293–302.
[41] L. Podsedkowski, J. Nowakowski, M. Idzikowski, and I. Vizvary. 2001. A new solution for path planning in partially known or unknown environment for nonholonomic mobile robots. *Robotics and Autonomous Systems* 34 (2001), 145–152.
[42] Z. Qu, J. Wang, and C. E. Plaisted. 2004. A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles. *IEEE Transactions on Robotics* 20, 6 (2004), 978–993.
[43] K. Sugihara and J. Smith. 1997. Genetic Algorithms for Adaptive Motion Planning of an Autonomous Mobile Robot. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*. IEEE, 138–143.
[44] O. Takahashi and R. J. Schilling. 1989. Motion planning in a plane using generalized Voronoi diagrams. *IEEE Trans. on Robotics and Automation* 5, 2 (1989), 143–150.
[45] Kartik Talamadupula, J. Benton, Subbarao Kambhampati, Paul Schermerhorn, and Matthias Scheutz. 2010. Planning for Human-robot Teaming in Open Worlds. *ACM Trans. Intell. Syst. Technol.* 1, 2 (Dec. 2010), 14:1–14:24.
[46] Y. Ting, W. I. Lei, and H. C. Jar. 2002. A path planning algorithm for industrial robots. *Computers and Industrial Engineering* 42 (2002), 299–308.
[47] Pratap Tokekar, Nikhil Karnad, and Volkan Isler. 2014. Energy-optimal trajectory planning for car-like robots. *Autonomous Robots* 37, 3 (Oct. 2014), 279–300.
[48] P. Vadakkepat, X. Peng, B. K. Quek, and Tong Heng Lee. 2007. Evolution of fuzzy behaviors for multi-robotic system. *Robotics and Autonomous Systems* 55, 2 (2007), 146–161.
[49] I. Škrjanc and G. Klančar. 2010. Optimal cooperative collision avoidance between multiple robots based on Bernstein-Bézier curves. *Robots and Autonomous Systems* 58, 1 (2010), 1–9.
[50] W. Wu, H. Chen, and P. Y. Woo. 2000. Time optimal path planning for a wheeled mobile robot. *Journal of Robotic Systems* 17, 11 (2000), 585–591.
[51] S. Xu, H. Wu, Y. Lin, H. Gao, and T. Chen. 2005. Path planning of mobile robot based on Voronoi diagram method. *Chinese Journal of Construction Machinery* 3, 3 (2005), 336–340.

[52] S. X. Yang and M. Meng. 2000. An efficient neural network method for real-time motion planning with safety consideration. *International Journal of Robotics and Autonomous Systems* 32 (2000), 115–128.

[53] W. Yuan, N. Ganganath, C. T. Cheng, G. Qing, and F. C. M. Lau. 2017. A consistent heuristic for efficient path planning on mobility maps. In *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 1–5.

[54] F. Benjamin Zhan and Charles E. Noon. 1993. Shortest path algorithms: An evaluation using real road networks. *Transportation Science* 32, 1 (1993), 65–73.

[55] B. Zhang, L. Tang, J. DeCastro, M. J. Roemer, and K. Goebel. 2015. A Recursive Receding Horizon Planning for Unmanned Vehicles. *IEEE Transactions on Industrial Electronics* 62, 5 (May 2015), 2912–2920.

Received July 2017; revised December 2017; accepted January 2018